Adaptive Wavelet Transforms Using Lifting Framework

A Thesis Submitted

in partial fulfillment of the requirements

for the Degree of

Master of Technology

by

TUSHAR PADLIKAR

to the

DEPARTMENT OF ELECTRICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY KANPUR

June 2004

28-6-04

CERTIFICATE

It is to certify that the work contained in this thesis entitled **"Adaptive Wavelet Transforms Using Lifting Framework"**, by Tushar Padlikar, Roll No. Y210442, has been carried out under my supervision and guidance. This work has not been submitted elsewhere for the award of any degree.

(Dr. Govind Sharma)
Professor
Department of Electrical Engineering
Indian Institute of Technology
Kanpur, India

IIT Kanpur
June, 2004

Abstract

In this thesis, a lifting based non-linear wavelet transform is proposed. This transform changes its filter support size adaptively depending on the data. A simple criterion for this adaptive change is proposed. The basic idea is to avoid using a large support size filter across the abrupt signal discontinuities(edges). This performs better then conventional linear algorithms. The reason being smaller number of wavelet coefficients is affected because of the edges. This gives an effective coding gain. The data itself is used for making the decisions, hence no side information needs to be sent along with the data. For coding the transformed image the SPIHT coder has been modified. The results of proposed transform are much better in case of synthetic test images and comparable in case of natural images. Future extensions to the algorithms have been proposed.

Acknowledgment

I would like to express gratitude and heartfelt thanks to my supervisor, Dr. Govind Sharma. It was because of his guidance, encouragement and valuable advice, that this thesis has taken a meaningful shape.

The word of gratitude should also go to all of my batchmates in ACES-205 laboratory for the technical discussions we had. In particular I would like to thank Ravi and Tushar for helping me with SPIHT coder and wavelet theory.

I would also like to thank all of my fellow students for making my stay in IIT Kanpur a memorable one.

Last but not least I would like to express my gratitude to wards my family without whom I would not have been what I am today.

Kanpur, India

June, 2004

(Tushar Padlikar)

Contents

List of Figures

Chapter 1

Introduction

Recent years have witnessed an explosion in the amount of information available in the form of digital image data. Take for instance, a database of images consisting of fingerprints/photographs in a law enforcement agency like Interpol. A 768×768 digital gray scale image will take at least 576 kilobytes of space at the rate of 8 bits per pixel(bpp). The number of images in such a database runs into millions and without any image compression algorithm the total database size goes well in terabytes. Though the storage/bandwidth constraints are surmounted to a great extent an efficient image compression algorithm always adds to the overall system performance. The field of Image/Video compression thus gains a special importance in all the branches of image processing. The following section serves as a general description of the existing methods.

1.1 A General Image Compression System

There have been many publications [12] attempting to find an optimal solution for the data compression problem. A general still image compression system has an architecture as shown in figure 1.1. Generally speaking, the image data is

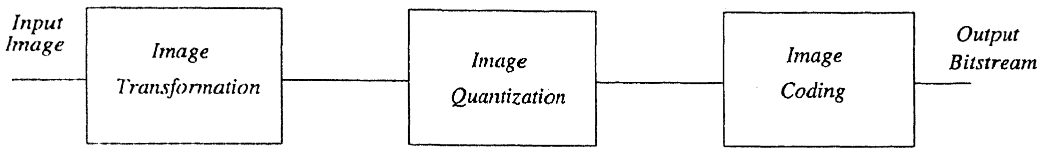

Figure 1.1: A General Image Compression System

transformed,depending on the type of compression desired(lossless/lossy) to some other domain, so as to separate the redundant and nonredundant parts of the data [13] [11]. These Transforms can be again be divided in two broad categories: Linear and Non-Linear transforms.

- **Linear Transform** : If $T(.)$ is a transformation acting on vectors x,y ,then it is called a linear transform iff,

$$T(ax + by) = aT(x) + bT(y)$$

  for all scalars a,b. Generally speaking, the significant properties of Linear transforms are:

  - Lower computational complexity then non-linear counterparts.
  - Lesser prone to errors,

  - The Linear Transforms are easier to deal with mathematically, concept of frequency analysis can be applied to these.

  DCT[1],DWT[2] fall into this category.

- **Non-Linear Transform** : The trivial definition of such transforms is the absence of linearity. Since these transforms do not have the constraints of the linear case there are additional degrees of freedom. These if properly exploited can be used to give such data representations which are better then linear counterparts for specific applications. The tradeoff is usually in terms of greater computational complexity and higher vulnerability to errors. Many approaches have been proposed in literature. The details can be found in [9] [10]

The transformed image is then quantized and subsequently coded in the form of a bit stream. The quantization step which is shown to be preceding, can be embedded in the coding step itself in some cases. If however, it is done as a separate stage the subbands are quantized in an individual manner. This makes sure that the perceptually important data has been coded at a higher bit rate. The coding scheme depends on the transformation used. SPIHT[3] is used after the image is transformed using DWT, refer section 4.3.1 on page 26 and a combination of RLE[4],Huffman and differential encoding schemes is used in case of DCT.

[1]DCT : Discrete Cosine Transform

[2]DWT : Discrete Wavelet Transform

[3]SPIHT : Set Partitioning in Hierarchical Trees

[4]RLE : Run Length Encoding

1.2 Overview of Thesis

In this work a non-linear transformation based on lifting framework [3],[6] is used. The additional degrees of freedom obtained by removing the constraint of linearity are used in changing the filter supports adaptively.

One of the problems of the DWT is the cone effect of the edges that is encountered at higher levels of decomposition [11]. The number of subband components that are affected by the abrupt signal discontinuities/edges goes on increasing. These subband coefficients are large in magnitude and hence it is not possible to code them efficiently. The basic idea in this thesis is to make sure that a minimum number of coefficients are affected because of the edges present in the images. This is done by adaptively detecting the presence of the edges and changing the filter support sizes in the proximity if the edges. When such introduction is made in the system the problems encountered include

- **Side Information overhead** The decisions about the filter usage done at the receiver should be the same as the ones that are made at the encoder. For this some side information should also be sent with the coded bit stream. This however is circumvented by using the coded data for making the filter choices

- **Synchronization** Since the side information is not being sent there should be a robust mechanism to make sure that the decision synchronism is maintained

- **Coding scheme** The coding scheme should be lossless. For this the conventional SPIHT coder has been modified

- **Error Vulnerability** Since the data transmitted is itself used for making the filter choices there should be robust error protection code for the same. This issue however, is not explored in this thesis.

The report is organized as follows. Chapter 2 discusses the theoretical aspects of the lifting scheme, dual lifting and gives an example of lazy wavelet. Chapter 3 extends the concept of lifting to adaptive lifting. The issues involved are elucidated in this chapter. Chapter 4 discusses the implementation issues involved. Detailed algorithms for one and two dimensional signals are given in this chapter. In Chapter 5 the results on some test images are discussed. The report concludes with Chapter 6 where possible extensions to this work are discussed.

Chapter 2

The Lifting Scheme and Second Generation wavelets

First generation wavelets have become a defacto standard for still image compression. Detailed analysis of first generation orthogonal wavelet representations can be found in [2], for biorthogonal settings refer [5]. First generation wavelets basis functions compact the energy of data in least number of coefficients possible. The reason being their ability exploit the spatio-frequency redundancy present in the data sets. These first generation wavelets are traditionally defined as dilates and translates of one particular $L_2(\mathbb{R})$ function, the mother wavelet $\psi(.)$,i.e, $\psi_{j,m} = \psi(2^j x - m)$.

The second generation [3] of wavelets on the other hand are not necessarily dilates and translates of a mother wavelet and are still able to enjoy all the powerful properties of first generation counterparts. Lifting scheme is a simple construction of second generation wavelets. Such wavelets can be adapted to intervals, domains, surfaces, weights, and irregular samples.

2.1 Definitions

- *Multiresolution analysis* : Let L_2 be a general function space of spatial domain with a weighted measure μ. Multiresolutuion analysis M is then defined as sequence of close supspaces: $M = \{V_j \subset L_2 \mid j \in J \subset \mathbb{Z}\}$, such that

  1. $V_j \subset V_{j+1}$,
  2. $\bigcup_{j \in J}$ is dense in L_2,
  3. for each $j \in J$, V_j has a Reisz bases given by scaling functions $\{\phi_{j,k} \mid k \in K(j)\}$

  $K(j)$ can be considered as a general index set. A dual multiresolution analysis $\widetilde{M}$ is defined in a similar manner with closed spaces $\widetilde{V}_j$ with Reisz bases scaling functions $\phi_{j,k}$. These dual scaling functions are biorthogonal with scaling functions in the sense that,

  $$\langle\ \phi_{j,k}, \widetilde{\phi}_{j,k'}\ \rangle = \ \delta_{k,k'} \qquad\qquad \textit{for } k, k' \in K(j)$$

- *Wavelets* : A set of functions $\{\ \phi_{j,m} \mid j \in J, m \in M(j)\ \}$, where $M(j) = K(j+1)/K(j)$ is a set of wavelet functions if

  1. The space $W_j = closspan\{\phi_{j,m} \mid \in M(j)\}$ is a complement of V_j in V_{j+1} and $W_j \perp \widetilde{V}_j$
  2. If $J = \mathbb{Z}$: The set $\{\psi_{j,m}/\|\psi_{j,m}\| \mid j \in J, m \in M(j)\}$ is a Reisz basis for L_2.

     If $J = \mathbb{N}$: The set $\{\psi_{j,m}/\|\psi_{j,m}\| \ \mid j \in J, m \in M(j)\} \cup \{\phi_{0,k}/\|\phi_{0,k}\| \mid k \in K(0)\}$ is a Reisz basis for L_2.

     Dual basis are given by $\widetilde{\psi}_{j,m}$, which are biorthogonal to the wavelets,

- *Refinement Relations* :

$$\phi_{j,k} = \sum_{l \in K(j+1)} h_{j,k,l}\phi_{j+1,l} \tag{2.1}$$

$$\psi_{j,k} = \sum_{l \in K(j+1)} g_{j,k,l}\phi_{j+1,l} \tag{2.2}$$

$$\phi_{j+1,l} = \sum_{k} \tilde{h}_{j,k,l}\phi_{j,k} + \sum_{m} \tilde{g}_{j,m,l}\psi_{j,m} \tag{2.3}$$

Where $h(.), g(.), \tilde{h}(.), \tilde{g}(.)$ are finite filters.

- *Biorhtogonality Conditions* :

  - On Scaling and Wavelet functions :

$$\langle\ \widetilde{\phi}_{j,k}, \phi_{j,k'}\ \rangle = \delta_{k,k'} \tag{2.4}$$

$$\langle\ \widetilde{\psi}_{j,m}, \psi_{j,m'}\ \rangle = \delta_{m,m'} \tag{2.5}$$

$$\langle\ \widetilde{\phi}_{j,k}, \psi_{j,m}\ \rangle = 0 \tag{2.6}$$

$$\langle\ \widetilde{\psi}_{j,m}, \phi_{j,k}\ \rangle = 0 \tag{2.7}$$

  - On Filters : Biorthogonality implies following relations for the filters

$$\sum_{l} g_{j,m,l}\widetilde{g}_{j,m',l} = \delta_{m,m'} \tag{2.8}$$

$$\sum_{l} h_{j,k,l}\widetilde{g}_{j,m,l} = 0 \tag{2.9}$$

$$\sum_{l} h_{j,k,l}\widetilde{h}_{j,k',l} = \delta_{k,k'} \tag{2.10}$$

$$\sum_{l} g_{j,m,l}\widetilde{h}_{j,k,l} = 0 \tag{2.11}$$

2.2 Lifting Scheme

Lifting scheme starts with a simple/trivial multiresolution analysis, and gradually is lifted up to a multiresolution analysis with particular properties and hence the

the name. As was pointed out earlier the lifting scheme is used for construction of second generation wavelets/filters.

Suppose one has access to a set of biorthogonal filters $\{h^{old}, \tilde{h}^{old}, g^{old}, \tilde{g}^{old}\}$ then a new set of biorthogonal filters can be constructed as following

$$\begin{aligned} h_{j,k,l} &= h^{old}_{j,k,l} && (2.12) \\ \tilde{h}_{j,k,l} &= \tilde{h}^{old}_{j,k,l} + \sum_m s_{j,k,m}\tilde{g}^{old}_{j,m,l} && (2.13) \\ g_{j,m,l} &= g^{old}_{j,m,l} - \sum_k s_{j,k,m}h^{old}_{j,k,l} && (2.14) \\ \tilde{g}_{j,m,l} &= \tilde{g}^{old}_{j,m,l} && (2.15) \end{aligned}$$

The effect of these changes are translated for scaling and wavelet functions as follows:

$$\begin{aligned} \phi_{j,k} &= \phi^{old}_{j,k} && (2.16) \\ \widetilde{\phi}_{j,k} &= \sum_l \widetilde{h}^{old}_{j,k,l}\widetilde{\phi}_{j+1,l} + \sum_m s_{j,k,m}\widetilde{\psi}_{j,m} && (2.17) \\ \psi_{j,m} &= \psi^{old}_{j,m} - \sum_k s_{j,k,m}\phi^{old}_{j,k} && (2.18) \\ \widetilde{\psi}_{j,m} &= \sum_l \widetilde{g}^{old}_{j,k,m}\widetilde{\phi}_{j+1,l} && (2.19) \end{aligned}$$

This is the essence of the lifting scheme. with operator S $\{s_{j,k}\}$ we have full control over all wavelets and dual functions that can be built from a particular set of scaling functions. This means we can start from a simple or trivial multiresolution analysis and choose S $\{s_{j,k}\}$ so that the wavelets after lifting have particular properties. This allows customdesign of the wavelet and it is the motivation behind the name "lifting scheme.". Conditions on $\psi_{j,m}$ thus immediately translate into

conditions on S. For example, S can be chosen so as to increase the number of vanishing moments of the wavelet, or such that $\psi_{j,m}$ resembles a particular shape. For the sake of example, consider a situation where the number of vanishing moments of the wavelets have to be increased.

For $\psi_{j,m}$ to have vanishing moments such that its integral multiplied to P_p is zero:

$$\int_X P_p \psi_{j,m} d\mu = 0 \Rightarrow \int_X P_p \psi_{j,m}^{old} d\mu = \sum_k s_{j,k,m} P_p \phi_{j,m}^{old} d\mu$$

For fixed indices j and m, this is a linear equation in unknowns $\{s_{j,k,m} \mid k \in K(j,m)\}$. Mathematical equations are realized in the filter bank framework [13] as shown in figure 2.1.

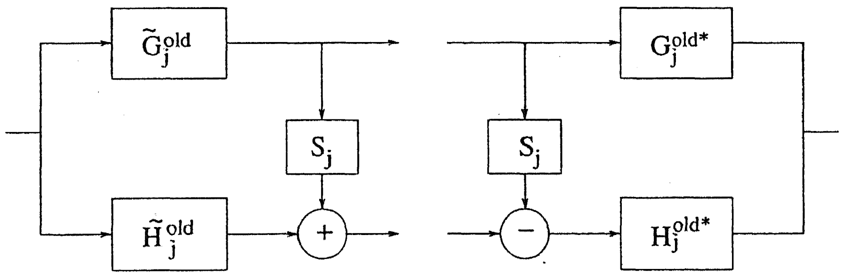

Figure 2.1: Lifting

2.3 Dual Lifting

Analogous to lifting is the concept of dual lifting. The basic idea is the same as for the lifting scheme except that now the dual scaling function is not changed. The equations for this case are given by

$$h_{j,k,l} = h_{j,k,l}^{old} + \sum_m \tilde{s}_{j,k,m} g_{j,m,l}^{old} \quad (2.20)$$

$$\tilde{h}_{j,k,l} = \tilde{h}_{j,k,l}^{old} \quad (2.21)$$

$$g_{j,m,l} = g^{old}_{j,m,l} \tag{2.22}$$

$$\tilde{g}_{j,m,l} = \tilde{g}^{old}_{j,m,l} - \sum_k \tilde{s}^*_{j,k,m} \tilde{h}^{old}_{j,k,l} \tag{2.23}$$

And as in the first case these conditions translate on the wavelet and scaling functions as:

$$\phi_{j,k} = \phi^{old}_{j,l} + \sum_m s_{j,k,m}\psi_{j,m} \tag{2.24}$$

$$\tilde{\phi}_{j,k} = \tilde{\phi}^{old}_{j,k} \tag{2.25}$$

$$\psi_{j,m} = \sum_m g_{j,m,l}\psi_{j,m} \tag{2.26}$$

$$\tilde{\psi}_{j,m} = \tilde{\psi}^{o}_{j,l}ld - \sum_k \tilde{s}^*_{j,k,m}\tilde{\phi}^{old}_{j,k,l} \tag{2.27}$$

The number of vanishing moments of the wavelet can be increased with the lifting scheme,and then the dual lifting scheme can be used to increase the number of vanishing moments of the dual wavelet. Desired properties on primal and dual wavelets can be achieved by iterating lifting and dual lifting alternately.

2.4 The Lazy Wavelet

In this section the Lazy wavelet, one of the initial multiresolution analysis to start with, is discussed. The Lazy wavelet transform is an orthogonal transform that resamples the coefficients into two groups at each step. That is, if $d(n)$ is the data sequence then it samples its odd and even coefficients out in $d_e(n)$ and $d_o(n)$.

A fast wavelet transform can be achieved by first applying a Lazy wavelet transform (resampling), then a dual lifting, and finally a regular lifting. The filter equations for such wavelet transform are then given by

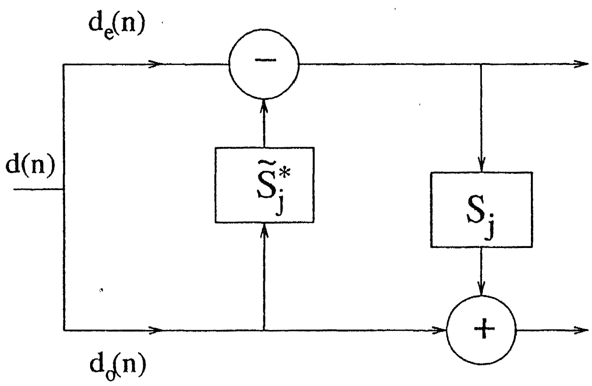

Figure 2.2: Lazy wavelet transform Dual Lifting and Regular Lifting

$$\widetilde{h}_{j,k,l} = \delta_{k,l} + \sum_m s_{j,k,m}\widetilde{g}_{j,m,l} \tag{2.28}$$

$$g_{j,m,l} = \delta_{m,l} + \sum_m s_{j,k,m}\widetilde{h}_{j,k,l} \tag{2.29}$$

2.5 Predict-Update Architecture

Previous sections have given introduction to the concept of second generation wavelets. Lifting scheme, however can be used to generate a multiresolution analysis for the case of first generation wavelets too. This section gives an introdcution to the notation that will be used in the report while referring to the lazy wavelet and subsequent lifting. The steps involved are:

- *Split*: In this step the samples of the data are resampled in odd and even components,
- *Predict*: Odd samples are predicted using even samples and a predict filter P. The Prediction error obtained is stored in the same spatial/temporal location

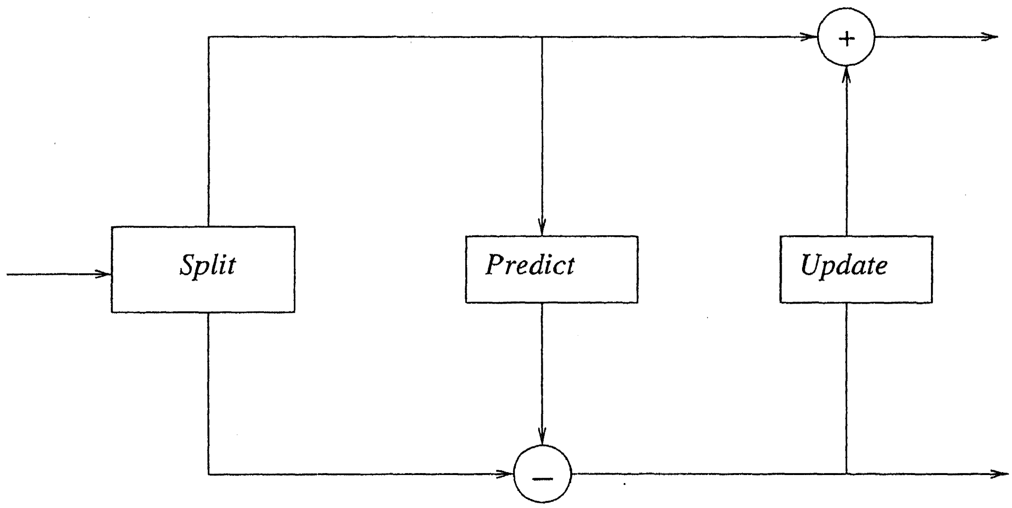

Figure 2.3: Predict-Update Architecture

as the odd sample predicted. This is same as the lifting step following the lazy wavelet transform

- *Update*: Prediction errors obtained in the previous step are used to update the even samples. This step is same as the dual lifting step following lazy wavelet transform and regular lifting.

Chapter 3

Adaptive Lifting

3.1 Introduction

The purpose of the transformation step in image coding is to represent the image efficiently with smaller number of coefficients. The wavelet transform provides such an efficient representation. Most of the wavelet coefficients of a typical image are nearly zero. The reason for the efficiency of the wavelet representation is that images often are well modeled as a set of locally smooth regions separated by edges. Within these smooth regions, fine-scale wavelet coefficients are small, and coefficients decay rapidly from coarse to fine scales. In the neighborhood of edges wavelet coefficients decay much more slowly, further the number of coefficients affected by the edges goes on increasing across the levels. These large wavelet coefficients near edges are expensive to code. A transform, which, in some sense, is edge sensitive, changing itself adaptively for transforming different regions of the image will be intuitively better then conventional transformation algorithms. Adaptive lifting is an extension to the the lifting framework for the wavelet transform. In adaptive

lifting a set of linear predictors (Section 3.3) that are chosen adaptively using a non-linear selection function. The following section elucidates the above idea in more specific terms.

3.2 Nonlinear Lifting

The analysis filters used for wavelet compression applications typically correspond to fourth order polynomial predictors which work well if the underlying signal is locally smooth. However, these predictions break down when the signal is not locally well-modeled as a low-order polynomial, i.e., near edges and other singularities. Using the lifting framework the algorithm switches between different predictors based on the local properties of the image. This gives a simple decision criterion, if the image is locally smooth, use higher order predictors and near edges use lower order predictor. In other words, for finding the wavelet coefficients on one side of the edge, near the edge, the length of the predictor is reduced to avoid using coefficients across the edge [1] This finally give us a non linear transform.

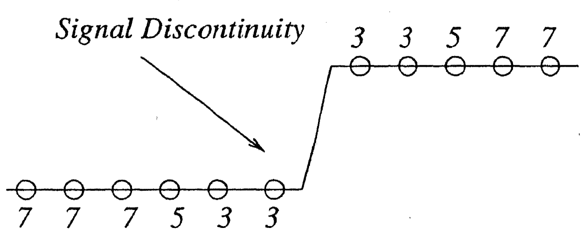

Figure 3.1: Figure showing change of filter support size in proximity of an edge

There are, however some problems associated while adding non-linearity to the lifting framework. Updated coefficient are interpreted as the low pass filtered

output in a conventional predict-update scheme(Section 2.5). To keep this interpretation as the low pass filter valid, update filter should be changed with the predict filters. One more problem can be serious reconstruction errors because of the restrictions imposed by the requirements of a conventional compression algorithm. The transformed image is quantized and thresholded so as to reduce as many coefficients to zero as possible. This however creates severe synchronization problems as the decisions of choosing the predictors are based on the data itself and if data is modified then the decisions made at the encoder and the decoder shall not be the same. Even if the problem of synchronization were solved by having the encoder to make its choices based on the quantized value there will still be a "loop" problem left. From the figure 3.2, it is observed that to maintain the synchronization predictor decisions have to made based on the quantized data $q\tilde{X}_e(n)$. But, for this data to be present at the encoder, we should have access to $\tilde{X}_o(n)$ which cannot be obtained without obtaining the information about the predictors. This leads to inconsistency and hence proves that conventional predict-update structure cannot be used.

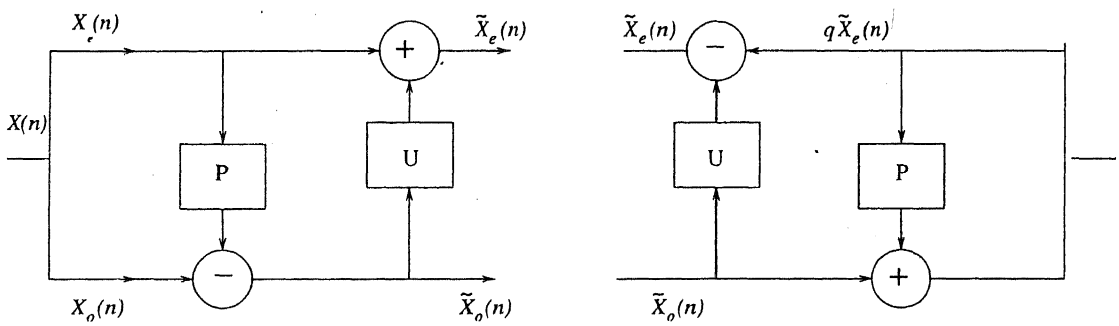

Figure 3.2: Update First Architecture

A solution to this loop problem is using update first architecture. The transform thus is carried out by first updating the even samples based on the odd samples

yielding the low pass coefficients. These low pass coefficients are reused to predict the odd samples, which gives the high pass coefficients. Low pass coefficients throughout the entire pyramid linearly depend on the data and are not affected by the non-linear predictor.

3.3 Choice of Filters

To fit the Update-Predict architecture, $(1, N)$ filters of Cohen-Daubechies-Feauveau family [5] are used. The filter coefficients for h and $\tilde{h}$ filters are obtained using following equations.

$$H(w) = \frac{(1+e^{-iw})}{2}^{N} e^{iw\lfloor N/2 \rfloor} \tag{3.1}$$

$$\tilde{H}(w) = (\cos w/2)^{\tilde{N}} \left[\sum_{n=0}^{k-1} (\sin w/2)^{2n} \right] \tag{3.2}$$

where $H(.)$ and $\tilde{H}(.)$ are the fourier transform of $h(.)$ and $\tilde{h}(.)$ filters. Further, N and $\tilde{N}$ are the vanishing moments of the wavelet and biorthogonal wavelet functions respectively and $2k = N + \tilde{N}$

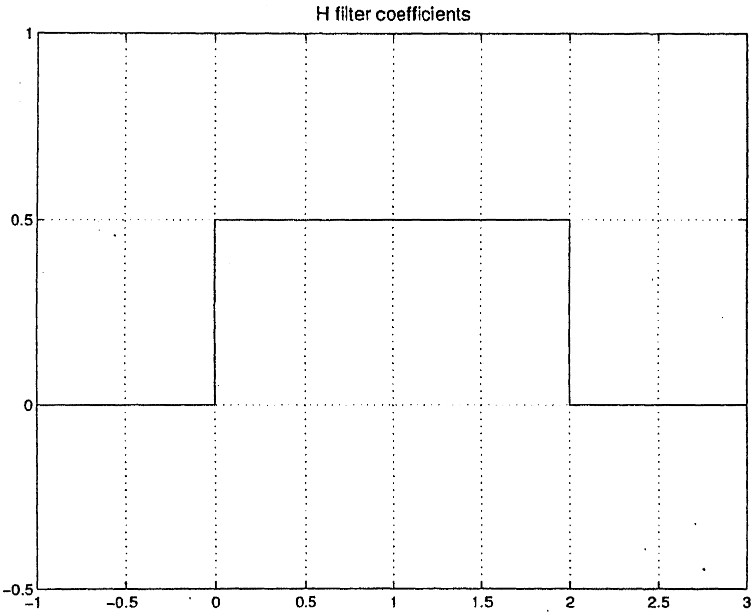

Figure 3.3: H filter

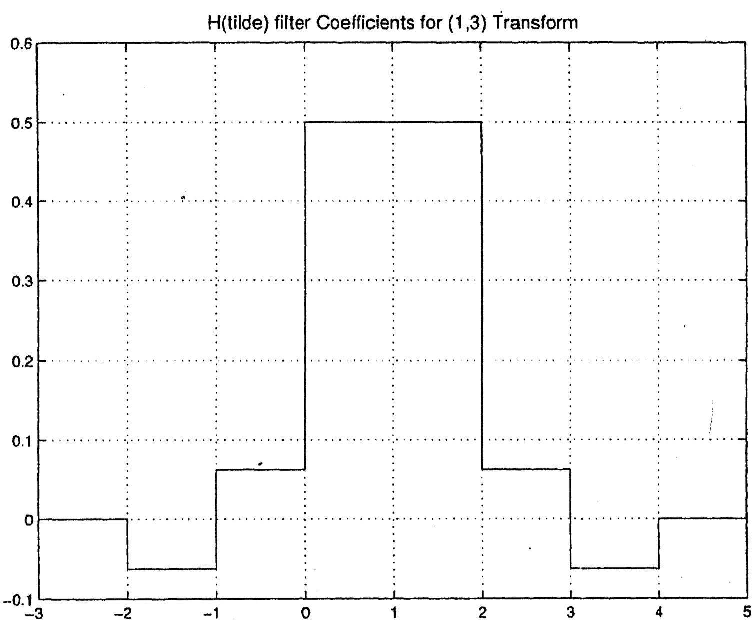

Figure 3.4: $\tilde{H}$ filter for (1,3) transform

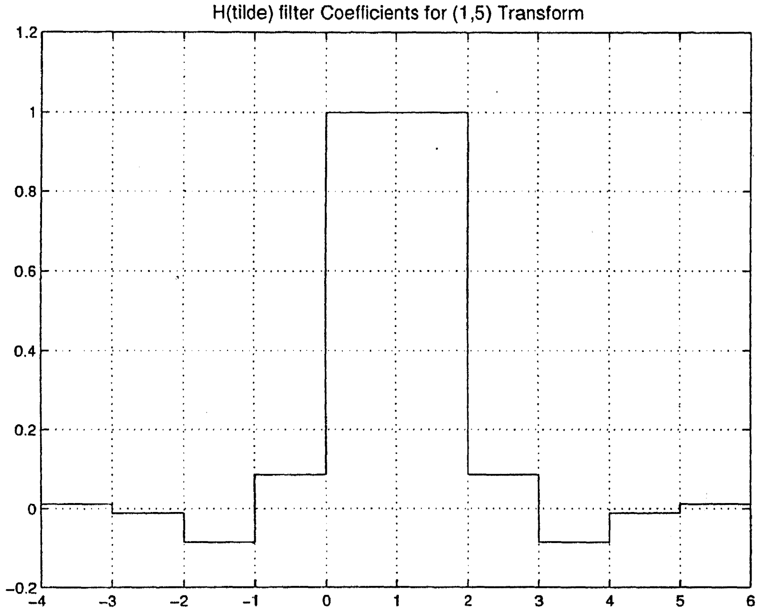

Figure 3.5: $\tilde{H}$ filter for (1,5) transform

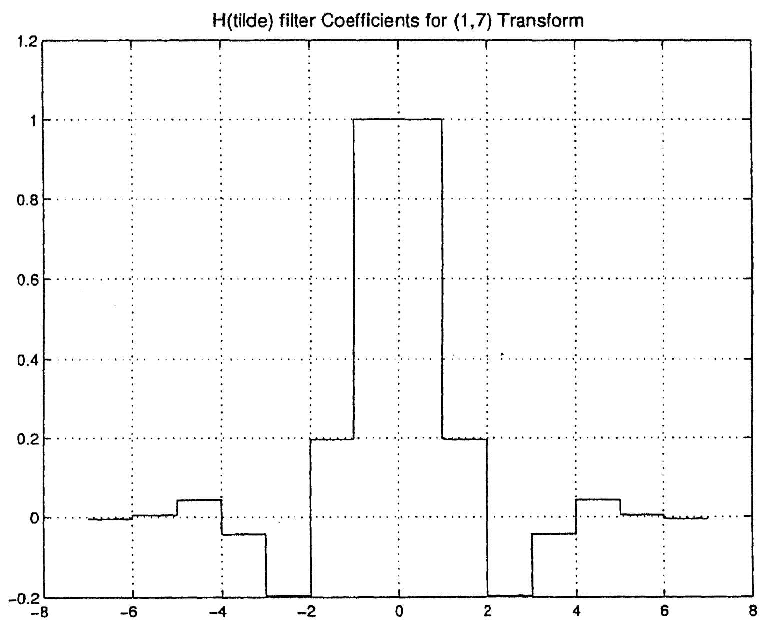

Figure 3.6: $\tilde{H}$ filter for (1,7) transform

The Predict filters ($P(n)$) used are :

- **(1,3)** $[\ -1/8 \quad 1 \quad 1/8\ ]$
- **(1,5)** $[\ -3/128 \quad 11/64 \quad 1 \quad -11/64 \quad 3/128\ ]$
- **(1,7)** $[\ 5/1024 \quad -44/1024 \quad 201/1024 \quad 1 \quad -201/1024 \quad 44/1024 \quad -5/1024\ ]$

with Update filter $U(n) = [\ 1/2 \quad 1/2\ ]$ in each one of the cases.

Chapter 4

Algorithm

This chapter discusses the algorithm used to implement the transformation. As was discussed in the previous section, the decisions about choosing the appropriate predictor from a set of predictors should be same at the encoder and the decoder to avoid the synchronization errors. To this end, it should be made sure that (a) the data at those spatial locations is available and (b) the data is same as it was at the encoder. Former one, seemingly trivial, has important repercussions.

- **Choice of predictors** depends on how well the data can be modeled by polynomials. To this end, a simple technique is devised. In the set of even data points (to be used for prediction) the even point closest to the odd point to be predicted, is predicted using the rest of the even samples. The error hence obtained is checked against some predefined threshold T . Depending on the error obtained, the predictors are chosen recursively. The predictor thus obtained is used for the final prediction and error storage. This way decisions depend on the even samples/scaling function projection coefficients, which in turn makes sure that the decisions at the encoder do not depend on

the stored prediction errors.

- To **maintain the synchronization** the predictor decisions are based on the quantized data, which implies that there has to be some feedback structure in the encoder. The need of this feedback structure can be further explained by looking at the figure 4.1.

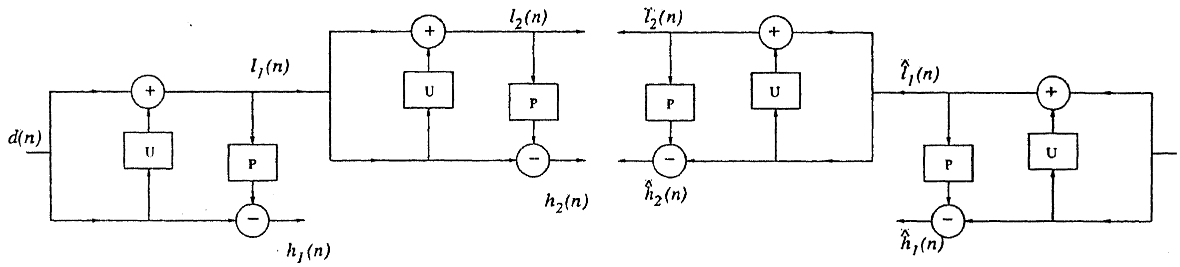

Figure 4.1: Figure showing inconsistency of Predict-Update approach

The figure shows a two level decomposition of a one dimensional signal $d(n)$.If the synchronization has to be maintained at the level two (for obtaining $h_2(n)$) the decisions should be based on the quantized $\hat{l}_2(n)$ rather than $l_2(n)$. Maintaining similar conditions at level one, however, needs access to data $\hat{l}_1(n)$. As is evident from the algorithm structure we need the data $\hat{h}_2(n)$ and $\hat{l}_2(n)$ for evaluating $\hat{l}_1(n)$. This explanation makes clear the necessity of having a feedback structure in the algorithm. The detailed algorithm for one dimensional decomposition is as follows.

4.1 One Dimensional Algorithm

The Pseudo-code for a one dimensional algorithm is as following:

One-Dim(Input Data $d(n)$,Level Decomposition n) {

For level $k = 1$ to n

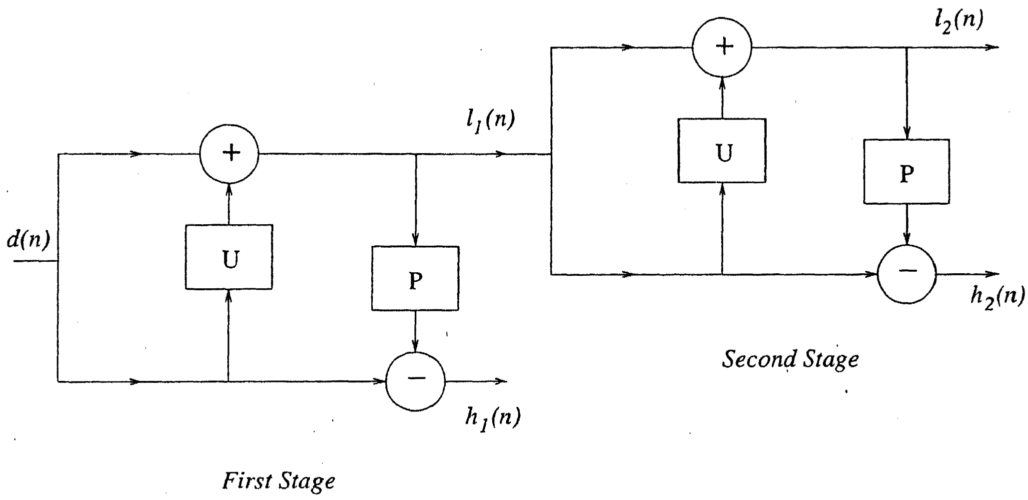

Figure 4.2: One Dimensional Algorithm

- Update the data

End

- Quantize the last level data
- Store in $Q(n)$

For level $k = n$ to 1

- Predict($Q(n)$,$l_k(n)$) $\rightarrow$ $h_k(n)$
- Quantize($h_k(n)$) $\rightarrow$ $\hat{h}_k(n)$
- if ($k \neq 1$) (if not at the first level)
  - One_ Step_ Reconstruction ($\hat{h}_k(n)$,$\hat{l}_k(n)$) $\rightarrow$ $\hat{l}_{k-1}(n)$
  - Store($\hat{l}_{k-1}(n)$) $\rightarrow$ $Q(n)$

End

}

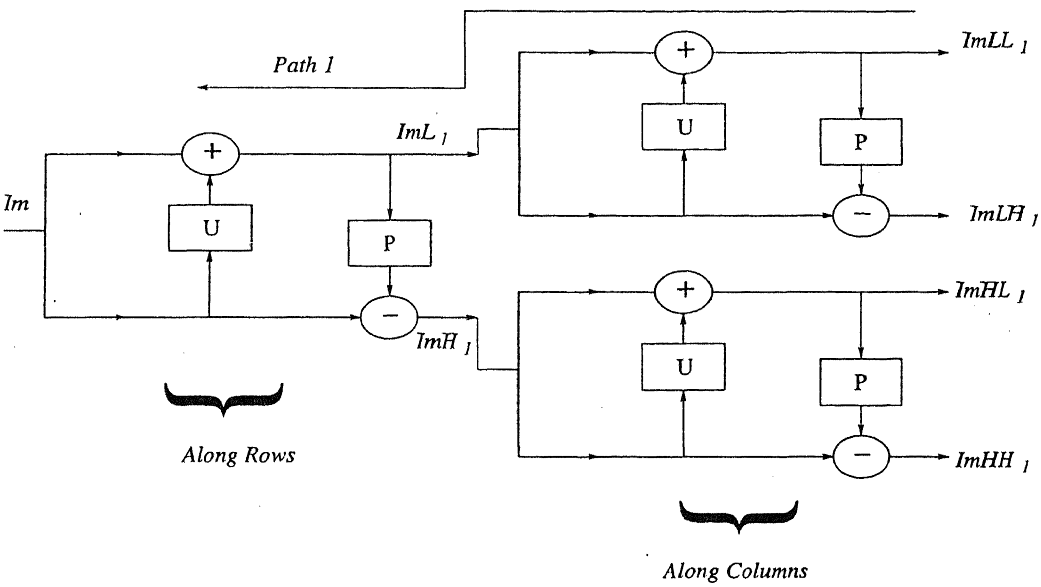

Figure 4.3: Two Dimensional Algorithm

4.2 Two Dimensional Algorithm

The filter in case of two dimensional data/images are considered separable, hence filtering is done along the rows and columns separately. The algorithm in case of images hence happens to be slightly different and is as explained below.

Two–Dim(Input Image Im , level decomposition n) {

For level $k = 1$ to n

- Update Image Along Rows
- Update Image Along Even Columns (Along Path 1)

End

- This gives $imLL_n$
- Quantize $(imLL_n) \rightarrow qimLL_n$

- Store $(qimLL_n) \to ImQ$

For level $k = n$ to 1

- Predict($ImQ, ImLL_k$) $\to ImLH_k$
- Quantize($ImLH_k$) $\to qImLH_k$
- One-step-reconstruction ($qImLL_k, qImLH_k$) $\to ImL_k$
- Predict ($qImL_k, ImL_k$) $\to ImH_k$
- Update-Along-Columns (ImH_k) $\to ImHL_k$
- Quantize($ImHL_k$) $\to qImHL_k$
- Predict ($qImHL_k, ImHL_k$) $\to ImHH_k$
- Quantize($ImHH_k)$) $\to qImHH_k$
- if ($k \neq 1$) (if not at the first level)
  - One-step-reconstruction ($qImHL_k)$ $,qImHH_k$) $\to qImH_k$
  - One-step-reconstruction ($qImL_k, qImH_k$) $\to ImL_{k-1}$
  - Store($qImL_{k-1}$) $\to ImQ$

End

}

4.3 Image Coding

The image is quantized in the transformation step itself. This quantization has to be preserved by any coding strategy to be used. In this work modified version of SPIHT(Set Partitioning In Hierarchical Trees) is used which is a well known algorithm used for coding images. This modification stems from the fact That the data is to be reconstructed perfectly from the bit stream the SPIHT generates. Following section gives a brief introduction of SPIHT, for details, refer [8]

4.3.1 SPIHT

The transformed image is considered to be present in the form of tiles(figure 4.4), and each coefficient is labeled $c_{i,j}$, where i, j are the pixel coordinates. Now, the algorithm selects the coefficients $c_{i,j}$ such that $2^n \leq |c_{i,j}| < 2^{n+1}$, with n decremented in each pass. Given n, if $|c_{i,j}| \geq 2^n$ then the coefficient is considered significant else it is called insignificant. The sorting algorithm partitions the coefficients in subsets T_m and performs magnitude tests.

$$max_{i,j \in T_m}\{|c_{i,j}\} \geq 2^n?$$

If the answer is no then all coefficients of that set are insignificant otherwise, using a rule shared by encoder and decoder the set T_m is partitioned into new subsets $T_{m,l}$, and the significance test is applied to the new subsets. This division process continues until the magnitude test is done to all single coordinate significant subsets in order to identify each significant coefficient.

Set partitioning rule should be such that subsets expected to be insignificant contain a large number of elements and the subset expected to be significant contains a single element. This requirement is satisfied by spatial orientation trees.

These spatial orientation trees exploit the spatial self similarity of the images by defining the nodes of the trees as points which are in same spatial locations in higher levels of subbands(figure 4.4). While partitioning the tree the two sub-

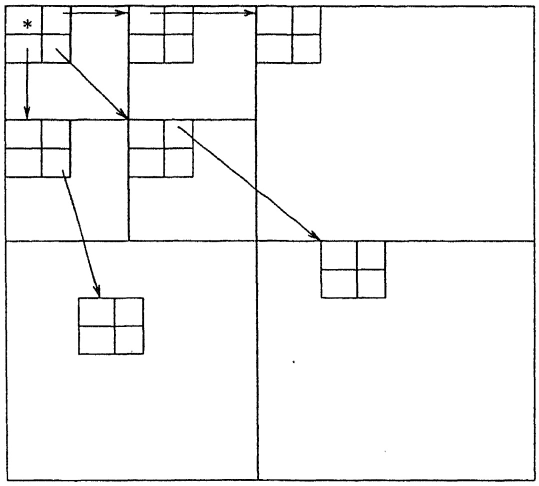

Figure 4.4: SPIHT tree structure

sets formed are of the immediate descendent's of the node and the rest of the descendants. Since a magnitude ordering is expected in the coefficients as one is traversing down the tree, this partitioning algorithm reduces the number of magnitude comparisons by exploiting spatial sef-similarity. Further, this partitioning rule is shared at the encoder and the decoder so the ordering information need not be sent with the data. Those coefficients which become significant in a certain pass are further refined by sending there bit planes. This algorithm terminates when the bit rate becomes equal to the one specified, but in this case since any quantization is not expected form the SPIHT. The sorting pass, thus terminates when n is such that 2^n is becomes less then the threshold defined while quantization and the refinement pass continues till all the bits of the coefficients that were candidates for significance, have been sent.

Chapter 5

Results and Discussions

The algorithm(Adaptive $(1, 7)$ transform) was used to transform and reconstruct test images at various bit rates. The PSNR was then compared with the following algorithms.

- *Non-Adaptive*$(1, 7)$ *transform*: This is essentially same as Adaptive$(1, 7)$ transform ,except for the adaptivity ,
- *Daubechies*$(9, 7)$ *transform*: The standard transform used in JPEG200, the wavelets and scaling functions have four vanishing moments each, The detailed mathematical properties of this transform are discussed in [7]

All the simulations were performed using Matlab, using the following test images :

- *Lena 512 $\times$ 512*
- *Synthetic image 512 $\times$ 512*

5.1 Results

5.1.1 Synthetic Image

The Adaptive(1,7) transform outperformed both the standard Daubechies(9,7) transform as well as Non-Adaptive(1,7) transform for synthetic images. The results are shown for three different bitrates. Figure 5.1 is the synthetic image on which the operation of decomposition and reconstruction was performed. As can be seen from the image, edges in all the directions are present. The reconstructed image after Adaptive(1,7) transform(figure 5.2) can be seen to preserve the edges in a better manner. This is attributed to the fact that a large number of wavelet coefficients are not effected because of the edges and after the quantization/thresholding operation on the transformed image, the edges do not spill over to neighboring areas as in the case of Daubechies(9,7) transform, see figure 5.4.

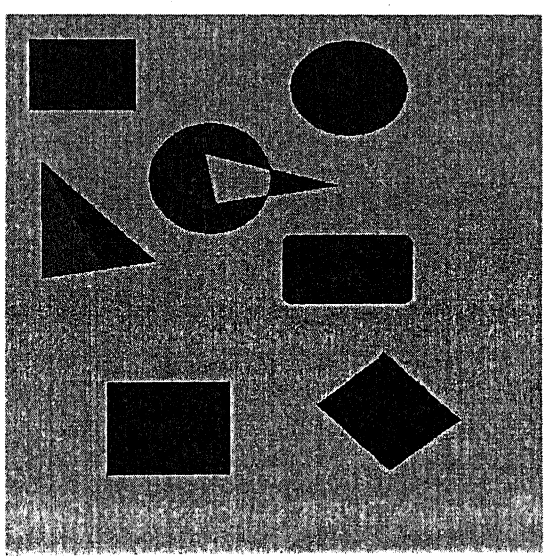

Figure 5.1: Original Synthetic Test Image

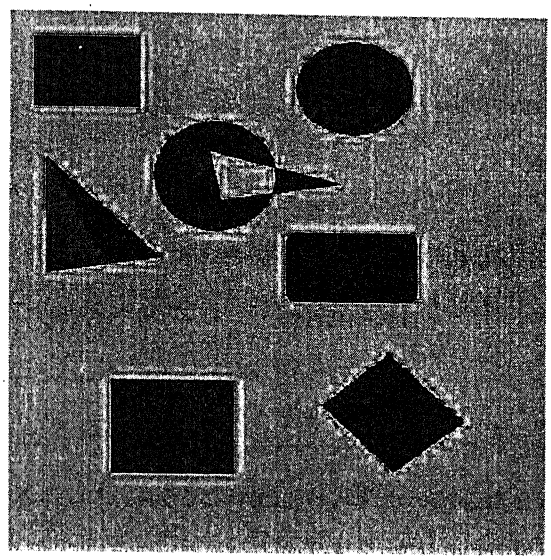

Figure 5.2: Reconstructed Image of Adaptive(1,7) Transform BPP 0.30 PSNR 27.974

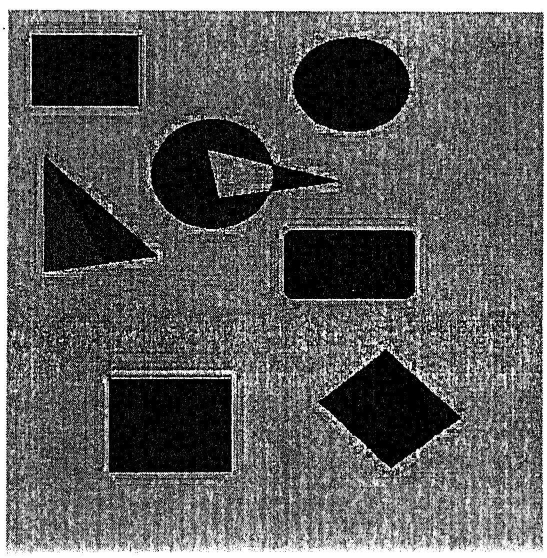

Figure 5.3: Reconstructed Image of Non-Adaptive(1,7) Transform BPP 0.30 PSNR 27.776

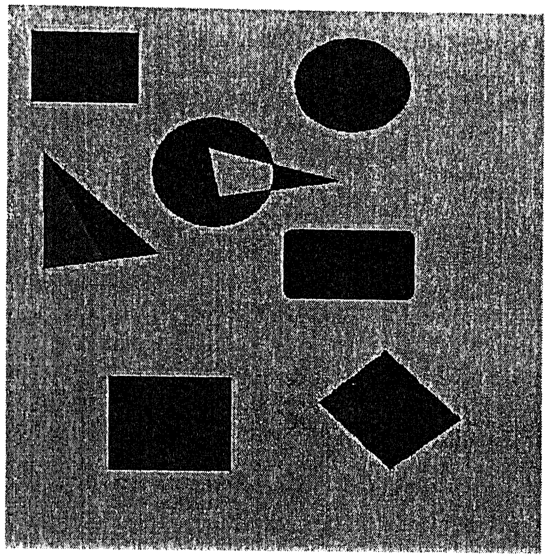

Figure 5.4: Reconstructed Image of Daubechies(9,7) Transform BPP 0.30 PSNR 20.832

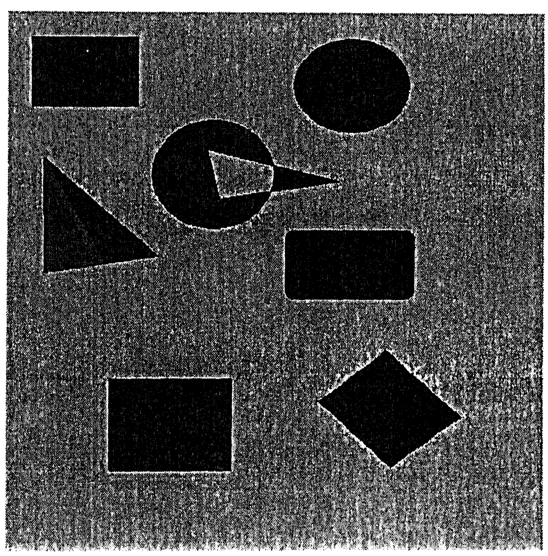

Figure 5.5: Reconstructed Image of Adaptive(1,7) Transform BPP 0.50 PSNR 39.526

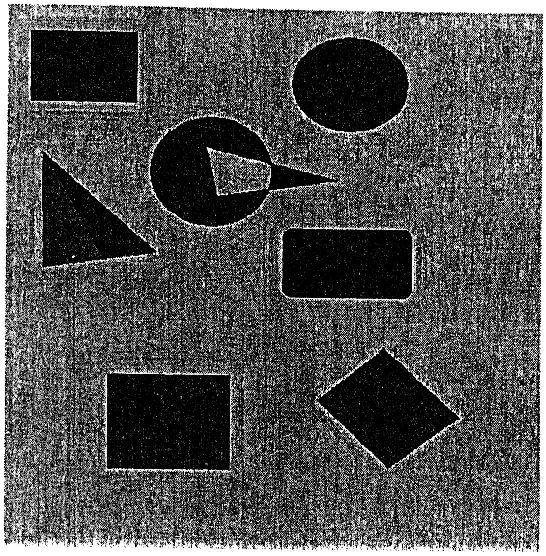

Figure 5.6: Reconstructed Image of Non-Adaptive(1,7) Transform BPP 0.50 PSNR 37.565

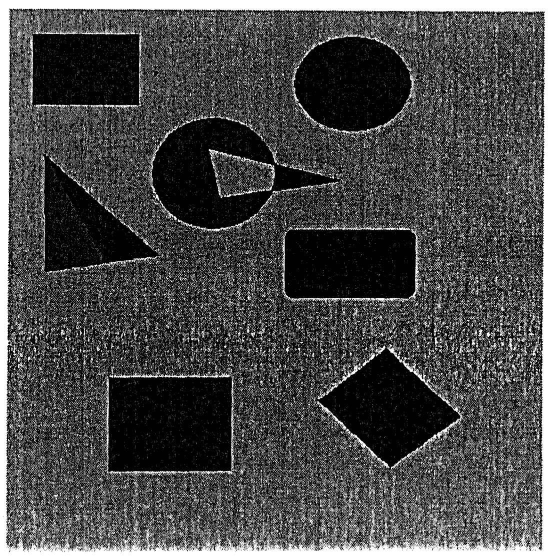

Figure 5.7: Reconstructed Image of Daubechies(9,7) Transform BPP 0.50 PSNR 32.693

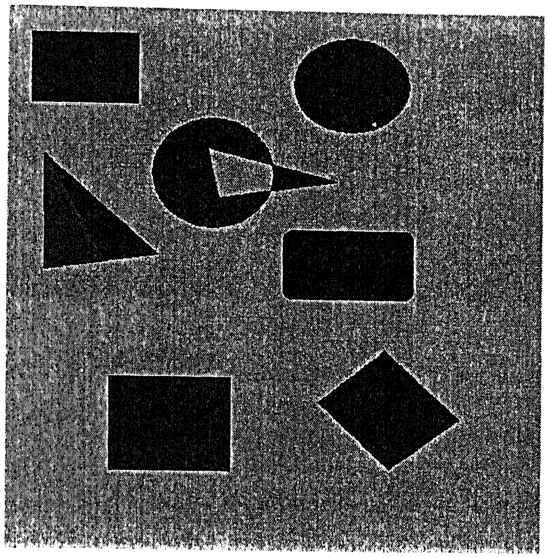

Figure 5.8: Reconstructed Image of Adaptive(1,7) Transform BPP 0.59 PSNR 47.496

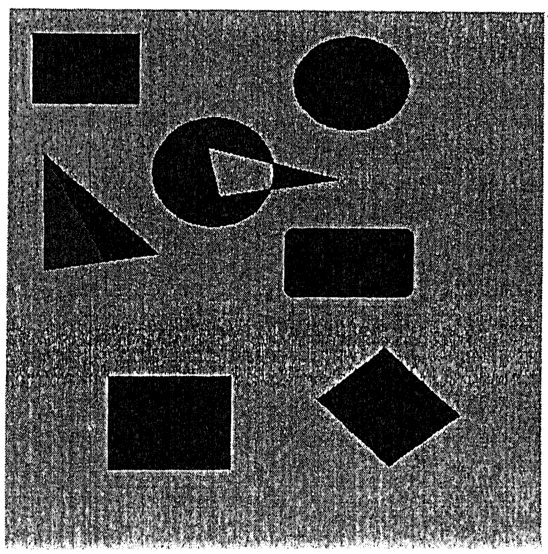

Figure 5.9: Reconstructed Image of Non-Adaptive(1,7) Transform BPP 0.60 PSNR 41.508

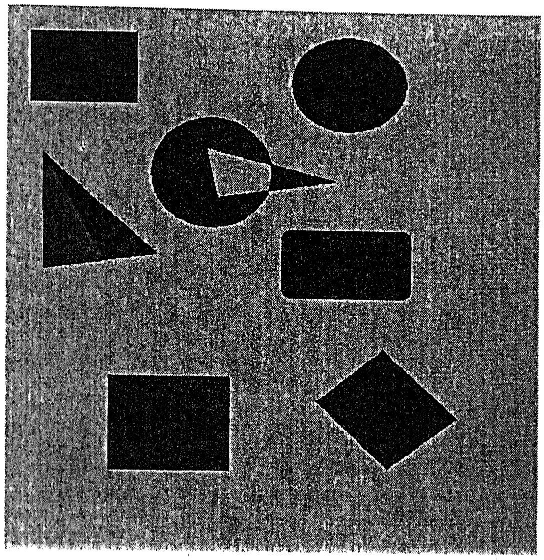

Figure 5.10: Reconstructed Image of Daubechies(9,7) Transform BPP 0.60 PSNR 37.379

5.1.2 Natural Image

Lena

The Transforms were also compared for the natural images. The results are presented here for standard gray scale lena image, figure 5.11. It is again seen that the performance of the adaptive transform,figure 5.12 is better then the other transforms(figure 5.14, figure 5.13). Here, though the PSNR performance of the proposed transform is always better, perceptually the two are comparable. This again validates the fact that PSNR performance does not always correspond to better perceptual appearance.

Figure 5.11: Original Lena Image

Figure 5.12: Reconstructed Image of Adaptive(1,7) Transform BPP 0.29 PSNR 28.008

Figure 5.13: Reconstructed Image of Non-Adaptive(1,7) Transform BPP 0.30 PSNR 23.442

Figure 5.14: Reconstructed Image of Daubechies(9,7) Transform BPP 0.30 PSNR 23.751

Figure 5.15: Reconstructed Image of Adaptive(1,7) Transform BPP 0.51 PSNR 31.229

Figure 5.16: Reconstructed Image of Non-Adaptive(1,7) Transform BPP 0.50 PSNR 26.228

Figure 5.17: Reconstructed Image of Daubechies(9,7) Transform BPP 0.50 PSNR 27.676

Figure 5.18: Reconstructed Image of Adaptive(1,7) Transform BPP 0.71 PSNR 32.297

Figure 5.19: Reconstructed Image of Non-Adaptive(1,7) Transform BPP 0.70 PSNR 28.026

Figure 5.20: Reconstructed Image of Daubechies(9,7) Transform BPP 0.70 PSNR 31.198

Comparisons

The plots of Bits per pixel versus PSNR for the three transforms are as shown in figures 5.21 and 5.22

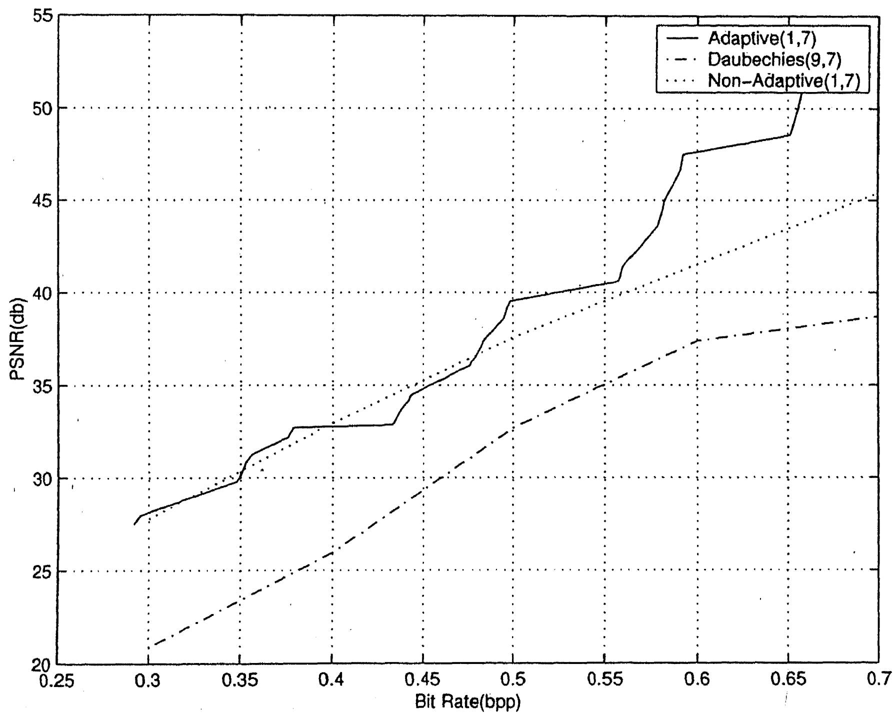

Figure 5.21: PSNR Vs BPP performance comparisons for Synthetic Image

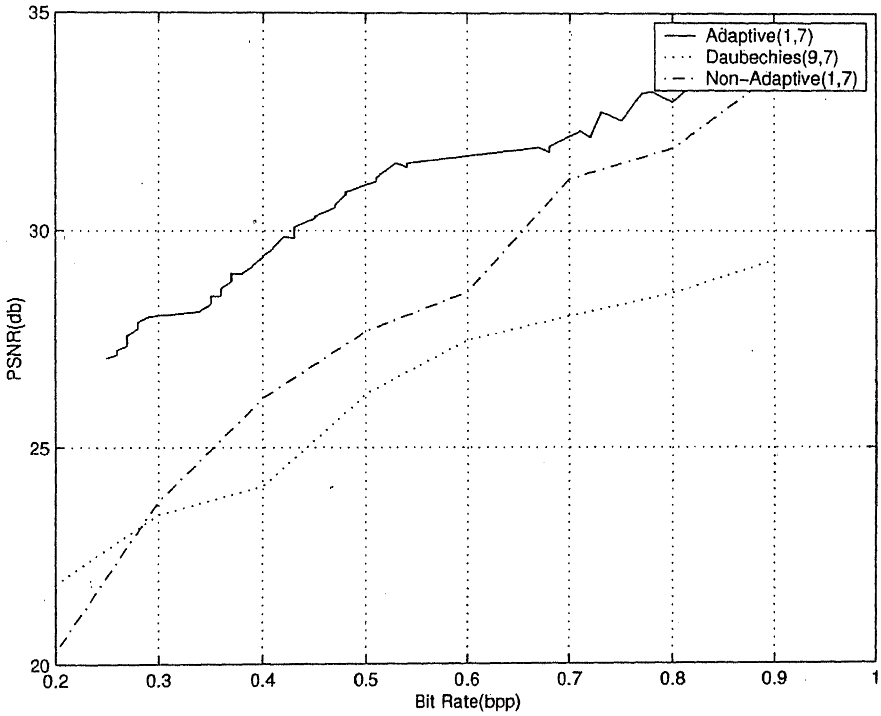

Figure 5.22: PSNR Vs BPP performance comparisons for Lena Image

Chapter 6

Conclusions and Future Scope

In this work the lifting framework was used for adaptive wavelet transforms. The Adaptive(1,7) found to perform better then Non-adaptive(1,7) transform in all the cases. It even outperformed daubechies(9,7) transform in case of synthetic images. In case of natural images, the performance of the Adaptive(1,7) is comparable to the standard daubechies(9,7) transform and as in the case of synthetic images, much better than Non-adaptive(1,7) transform. Proposed transform performs distinctively better near the edges. This is because of algorithms' ability to change the filter size which make sure that not a very large number of coefficients are affected because of the edges.

Future Scope

- In this work the adaptivity was added to a (1,7) transform. If this can be extended to Daubechies(9,7) transform the results can be expected to be better.

- The idea of adaptive lifting can be extended to wavelet packet decomposition
- The idea can be extended to 3-D video coding.
- The bit stream if subjected to errors severely degrades the image. This is because these errors propagate across the scaled in form of incorrect filter decisions. Some robust error protection should be used

Bibliography

[1] R. Claypoole, G. Davis, W. Sweldens and R. Baraniuk,"Nonlinear wavelet transforms for image coding via lifting" ,*IEEE Transactions on Image Processing*,Vol 12, No 12 pp 1449-1559,December 2003

[2] Ingrid Daubechies,*Ten lectures on wavelets.* PA:Society for Industrial and Applied Mathematics, 1992

[3] W. Sweldens, "The lifting scheme: A construction of second generation wavelets". *SIAM J. Math. Anal.*, 1997.

[4] W. Sweldens and P. Schroder,"Building your own wavelets at home". *Wavelets in Computer Graphics*, pages

[5] A. Cohen, I. Daubechies, and J. Feauveau,"Bi-orthogonal bases of compactly supported wavelets" *Comm.Pure Appl. Math.*, 45:485–560, 1992.

[6] W. Sweldens, "The lifting scheme: A custom-design construction of biorthogonal wavelets".*Appl. Comput.Harmon. Anal.*, 3(2):186–200, 1996.

[7] Michael Unser et al., "Mathematical Properties of JPEG200 wavelet filters".*IEEE Transaction on Image Processing* vol. 12 no. 9 September 2003

[8] A. Said , A pearlman,"A new, fast, and efficient image codec based on set partitioning in hierarchical trees", *IEEE Transactions on Circuits and Systems for Video technology* Vol. 6 No. 3 June 1996

[9] Heijmans et al,"Nonlinear multiresolution signal decomposition schemes. II. Morphological wavelets",*IEEE Transactions on Image Processing*,Vol 9, No 11 pp 1897-1913 ,November 2000

[10] F.J. Hampson et al,"M-band nonlinear subband decompositions with perfect reconstruction",*IEEE Transactions on Image Processing*,Vol 7, No 11 pp 1547-1560,November 1998

[11] S. Mallat, *A Wavelet Tour of Signal Processing.* New York:Academic,1998

[12] A. Bovik, *Handbook of Image and Video Processing.*Canada:Academic,2002

[13] N.J. Fliege,*Multirate Digital Signal Processing*, Chichester,UK:John Wiley & Sons, 1993